

Scalable Hybrid Verification for Embedded Software

Behrend, J.; Lettnin, D.; Heckeler, P.; Ruf, J.; Kropf, T.; Rosenstiel, W. *"Scalable Hybrid Verification for Embedded Software"*. Design, Automation & Test in Europe Conference & Exhibition (DATE), 2011 , vol., no., pp.1-6, 14-18 March 2011

Aluno: Junior John Fabian Arteaga

RA: 123542

Atualmente, o software embarcado está presente em nossa vida diária. Este tipo de software é usado para controlar as funções do carro, produtos de telecomunicação, robôs, aparelhos elétricos, dispositivos médicos, etc. O software embarcado desempenha um papel fundamental com funcionalidades como a redução das emissões de combustíveis, a melhoria da segurança entre outras.

O poder de processamento cada vez maior, a diminuição do consumo de energia, a diminuição dos preços e a minimização do tamanho dos microcontroladores e microprocessadores permitem desenvolver mais funcionalidades do hardware no software embarcado; o qual termina com milhões de linhas de código e um rápido aumento da complexidade.

A maioria dos erros fatais ocorre nos softwares dependentes do hardware e não nos software de aplicação. Alguns exemplos de diferentes erros em software dependente de hardware (e.g., drivers de baixo nível) são os erros de temporização, estouro da memória, erros de inconsistência. É por isso que, a verificação de software embarcado tornou-se um tema importante nos últimos anos.

Este trabalho apresenta uma nova abordagem híbrida e escalável para a verificação de propriedades temporais em softwares embarcados com dependências de hardware. O método de verificação proposto consiste de três principais fases: Preprocessamento, "Bottom-up" e "Top-down". No processamento, o código em C é processado mudando o código para 3-AC (three-address code) que é normalmente usado por compiladores. Seguidamente, geram um modelo "SystemC" do software embarcado. Depois na etapa de Bottom-up, verificam as principais funções do software e marcam as funções segundo um modelo verificador presente no estado da arte. Y finalmente na etapa Top-down, analisam as funções marcadas na fase anterior. Os autores apresentam dois tipos de experimentos diferentes nos quais mostram que a abordagem de verificação proposta tem bons resultados e é facilmente utilizado para softwares embarcados complexos dependentes do hardware. Além disso, a abordagem é mais escalável em comparação com métodos propostos no estado da arte.